Digital India RISC-V VEGA Processor

(Fully indigenous RISC-V Processor)

DIR-V VEGA Processor by C-DAC

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

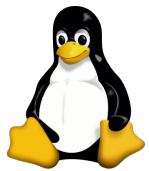
Installing VEGA SDK

DIR-V VEGA Processor by C-DAC

Installing VEGA SDK

What you will learn?

- Install vega-tools
- Install vega-sdk
- Building a project
- Running Program on Target Board
- Creating a New Project
- Interrupt-Based Example
- Using SDK in IDE (Eclipse)
- Executing the Program on Target Board via Ethernet

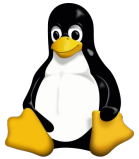


DIR-VEGA Processor | C-DAC

Installing VEGA SDK

Supported Systems

- Linux machines only
- Ubuntu



DIR-V VEGA Processor by C-DAC

Supported Systems

Currently the Linux distributions that support VEGA SDK are :

- Ubuntu 18.04
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04

- We expect other Linux distributions to work as well, provided that the user can either run our provided toolchains on them or produce their own toolchain.

DIR-V VEGA Processor by C-DAC

Prerequisites

To use this SDK, you will need the following software available on your Linux machine :

GNU Make

- `sudo apt update`
- `sudo apt install build-essential`

GNU Autoconf

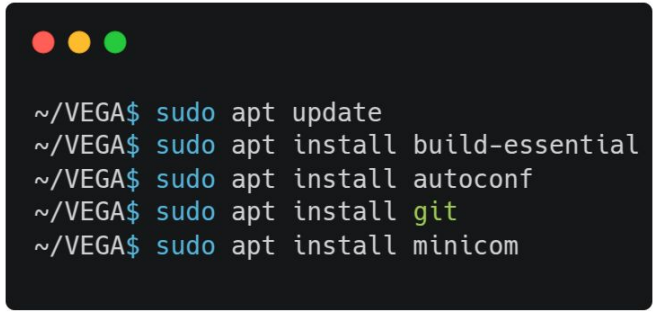
- `sudo apt install autoconf`

Git

- `sudo apt install git`

Minicom

- `sudo apt install minicom`



DIR-V VEGA Process by C-DAC

Install VEGA Tools

- VEGA Tools contains necessary RISC-V Toolchains to compile C/C++ programs and other utilities.
- To Install VEGA Tools :

For TRISUL32 (Arty A7-100T)

- `git clone https://gitlab.com/dirv-vega-gc2025/vega-tools.git`
- `cd vega-tools`
- `./setup-env.sh`
- `cd ..`

DIR-V VEGA Processor by C-DAC

Install VEGA SDK

Clone and install VEGA SDK

- `git clone https://gitlab.com/dirv-vega-gc2025/vega-sdk.git`
- `cd vega-sdk`

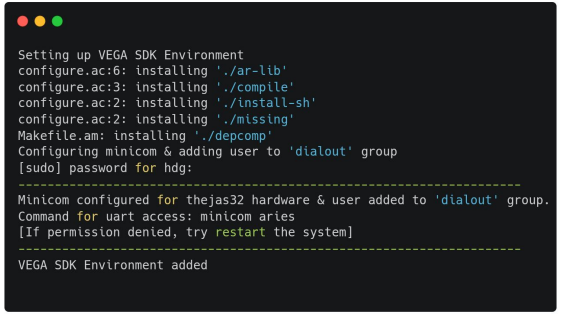
Install VEGA SDK(contd...)

VEGA SDK supports several development boards.
Choose the setup script according to your target development board.

For TRISUL32 (Arty A7-100T)

- `./setup-trisul32.sh`

If all steps are successful, you'll see the following output after running `./setup-trisul32.sh`

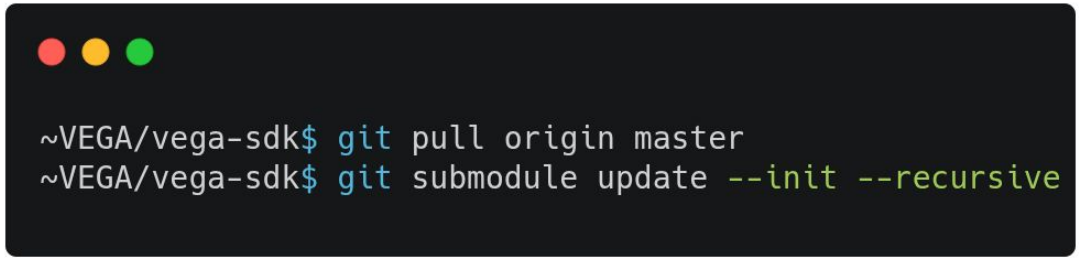


DIP-V VEGA Processor by C-DAC

Updating the SDK/Tools

If you'd like to update your SDK/Tools to the latest version, go to respective directory and :

- `git pull origin master`
- `git submodule update --init --recursive`



Note

- It is recommended to update VEGA Tools before you update VEGA SDK.

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Building a Project

DIR-V VEGA Processor by C-DAC

Building a Project

To compile an example project, say hello_world

- `cd examples/uart/hello_world/`
- `make`

```
~/VEGA/vega-sdk$ cd examples/uart/hello_world/
~/VEGA/vega-sdk/examples/uart/hello_world$ make
```

If your make command end like following,

```
Build successful!
ELF : hello
Binary  : hello.bin
Hex : hello.hex
Dump    : hello.dump
Files are generated in build folder.
Size information
   text    data     bss     dec     hex filename
  26072      36      84   26192    6650 build/hello.elf
```

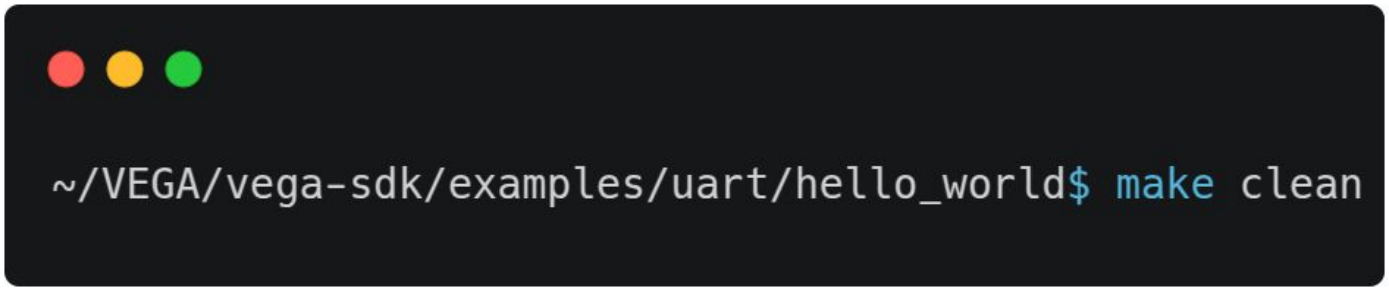
- Then your build is successful.
- After a successful build, an **elf** file, **.bin** file, **.dump** file and a **.hex** file will be created in the **build/** directory.
- **.dump** file contains your program in assembly form.

Cleaning a Target Project Build Directory

The clean target can be used to restore a target project's directory to a clean state.

- `make clean`

This will remove the **build/ directory** of the target project.



DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Running Program on Target Board

DIR-V VEGA Processor by C-DAC

Running Program on Target Board

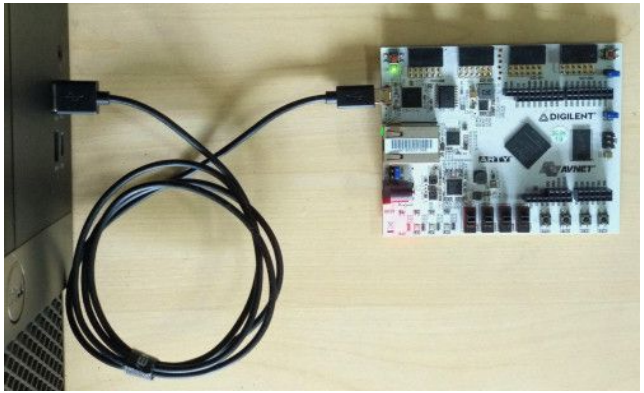
Setting Up Serial Device

- We are using Linux application minicom for serial communication.
- You could use any other serial communication applications like **python-serial**.

Note

- Make sure that the board is connected to the host PC via micro USB cable.
- Also ensure that you use the correct serial device! In most cases it would be `/dev/ttyUSB1`.

- `minicom -h` will give you an idea of options available in minicom.

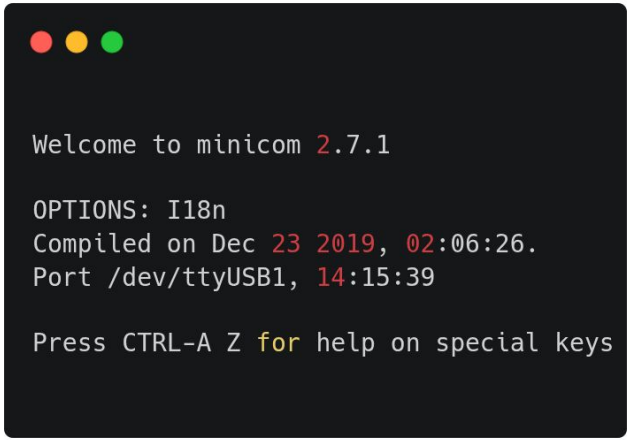


DIR-V VEGA Processor by C-DAC

Running Program on Target Board

To setup minicom :

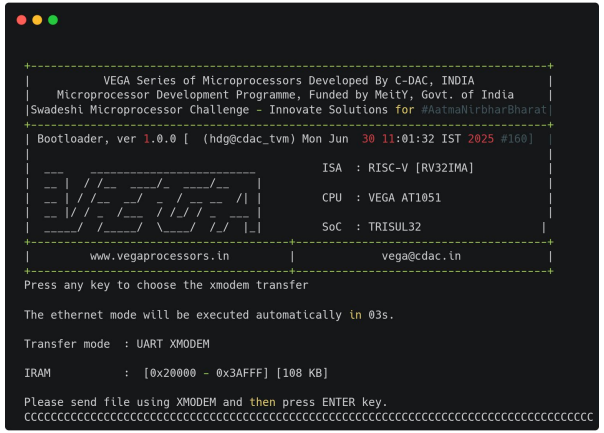
- `sudo minicom trisul32`



DIR-V VEGA Processor

Resetting Target Board

- You could reset target board by pressing the reset button.
- You should reset the Board every time before you upload programs to the Board.
- You could find reset button of your target board from Board pin mapping.
- If your reset was correct you will see something like this in terminal :



Uploading Program to Target Board

- While uploading, we are transferring the **.bin** file to the target board and executing that file on target board.
- VEGA SDK supports many methods for uploading program.
- Follow the method supported by your target board.

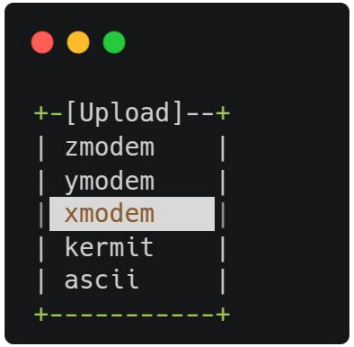
TRISUL32 (Arty 100T)	Uploading Using UART
	Uploading Using Ethernet

DIR-V VEGA Processor by C-DAC

Uploading Using UART

Uploading using UART method uses XMODEM protocol to upload program.

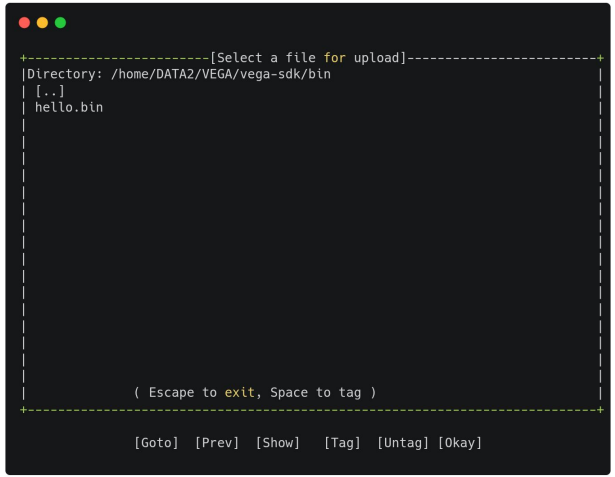
- Open Minicom and press the RESET button to restart the board.
- Press any key to choose the xmodem transfer and make sure Transfer mode is UART XMODEM.
- Use `CTRL+A S` to enter file sending menu and select xmodem by pressing Enter.

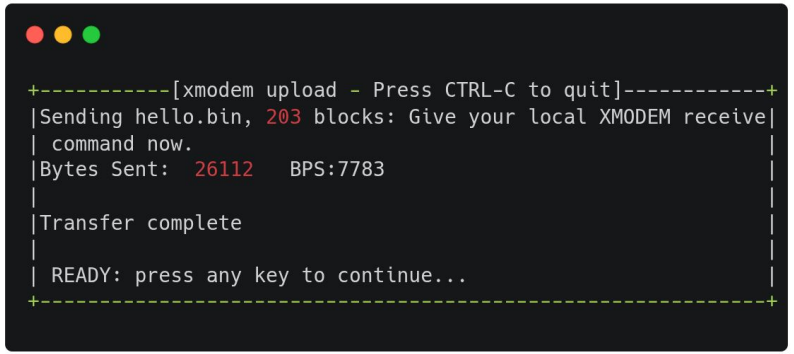


DIR-V V... processor by C-DAC

Uploading Using UART (contd...)

- In the next window, with space bar select the `.bin` file to be transferred, by pressing Enter, transfer process starts.
- Wait until the process is completed. The screen should display how much data has been transferred.





Uploading Using UART *(contd...)*

- After completing transfer, the Program will start to execute.

```
Transfer mode  : UART XMODEM

IRAM           :  [0x20000 - 0x3AFFF] [108 KB]

Please send file using XMODEM and then press ENTER key.
CCCCCCCCC

Hello World
EXIT
```

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Creating and Uploading a New Project to the Target Board

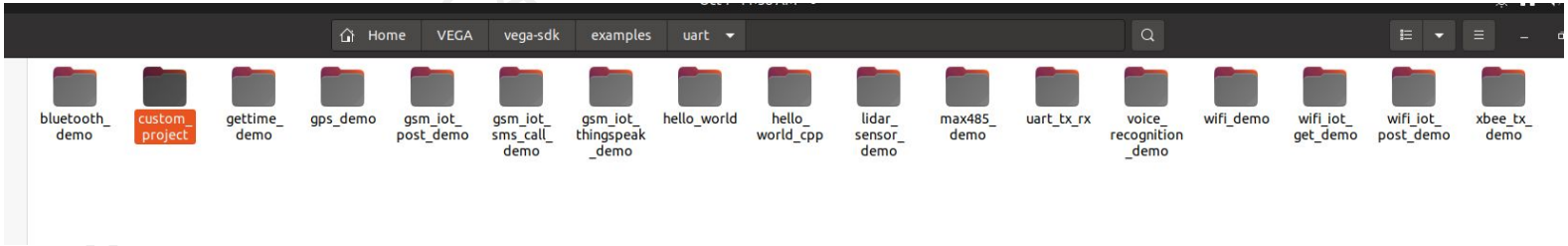
DIR-V VEGA Processor by C-DAC

Creating a Project

It is recommended to create a new project by copying any sample projects from **examples/** directory.
You can also make a new Project by just copying the MakeFile only.

Recommended Method :

- Copy any example project and rename, say `custom_project.`



Creating a Project(contd...)

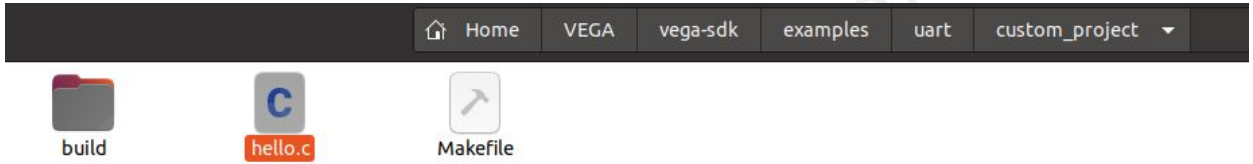
- Change the `EXECUTABLE_NAME` variable in the MakeFile to `EXECUTABLE_NAME= custom_project.`

```
#-------------------------------------------------------------------------
#Project Name          : MDP - Microprocessor Development Project
#Project Code          : HD083D
#Created               : 07-Jan-2020
#Filename              : Makefile
#Purpose               : Sample hello world program
#Description           : Sample hello program with uart print
#Author(s)             : Premjith A V
#Email                 : premjith@cdac.in
#-------------------------------------------------------------------------
#See LICENSE for license details.

#++++++++++++++++++++++++
# Configurations
#++++++++++++++++++++++++
# Include the BSP settings

CONFIG_PATH=~/.config/vega-tools/settings.mk
ifeq ("$(wildcard $(CONFIG_PATH))","")
$(error Please install [VEGA SDK]/[VEGA Tools] and setup the environment)
endif

include $(CONFIG_PATH)

ifeq ("$(wildcard $(VEGA_TOOLCHAIN_PATH))","")
$(error Please install [VEGA Tools] and setup the environment)
endif

ifeq ("$(wildcard $(VEGA_SDK))","")
$(error Please install [VEGA SDK] and setup the environment)
endif
SDK_PATH=${VEGA_SDK}

#++++++++++++++++++++++++
# Executable name
#++++++++++++++++++++++++
EXECUTABLE_NAME=custom_project


include $(SDK_PATH)/bsp/common/config.mk

```

DIR-V VEGA Proc

Creating a Project(contd...)

- Remove all `.c` and `.h` files in `custom_project`.



DIR-V ... by C-DAC

Creating a Project(contd...)

- Create your own `.c` file say, `custom_project.c` and implement your logic.

Open — custom_project.c
~/VEGA/vega-sdk/examples/uart/custom_project

```c
/*****************************************************************************

 Project Name          : MDP - Microprocessor Development Project
 Project Code          : HD083D
 Created               : 07-Jan-2020
 Filename              : hello.c
 Purpose               : Sample hello world program
 Description           : Sample hello program with uart print
 Author(s)             : Premjith A V
 Email                 : premjith@cdac.in

 See LICENSE for license details.
*****************************************************************************/

#include "stdlib.h"
/**
 @file hello.c
 @brief Contains routines for Prints a sample text in console.
 */


 /**
  * Hello World program main function
  * Prints a sample text in console.
  */

int main ()
{
        printf("\nC-DAC VEGA Processors\n");
}
```

- Use `make` command to build your project.

```
~/VEGA/vega-sdk/examples/uart/hello_world$ cd ..
~/VEGA/vega-sdk/examples/uart$ cd custom_project/
~/VEGA/vega-sdk/examples/uart/custom_project$ make
```

DIRV VEGA Pro

Creating a Project(contd...)

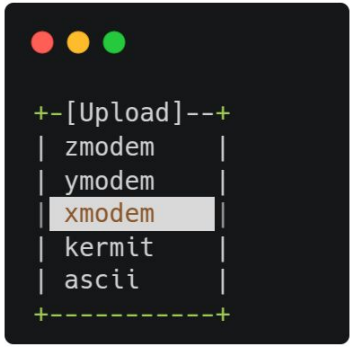
- If your make command end like following,Then your build is successful.

```
Build successful!
ELF : custom_project
Binary  : custom_project.bin
Hex : custom_project.hex
Dump    : custom_project.dump
Files are generated in build folder.
Size information
   text    data     bss     dec     hex filename
  26812      36      84   26932    6934 build/custom_project.elf
```

Uploading Using UART

Uploading using UART method uses XMODEM protocol to upload program.

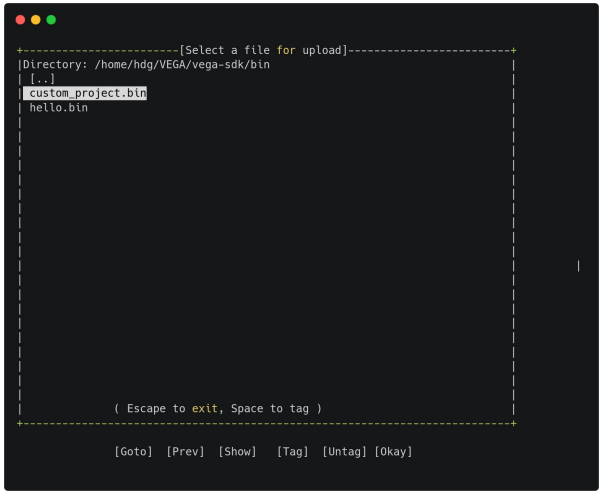
- Open Minicom and press the **RESET** button to restart the board.
- Press any key to choose the xmodem transfer and make sure Transfer mode is UART XMODEM.
- Use **`CTRL+A S`** to enter file sending menu and select xmodem by pressing Enter.

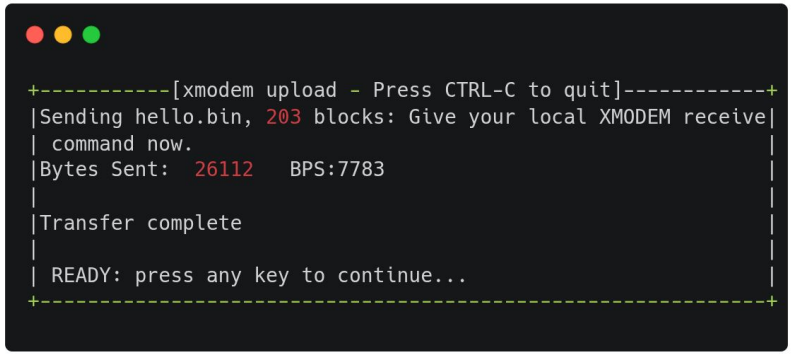


DIR-V V Processor by C-DAC

Uploading Using UART (contd...)

- In the next window, with space bar select the `.bin` file to be transferred, by pressing Enter, transfer process starts.
- Wait until the process is completed. The screen should display how much data has been transferred.





Uploading Using UART *(contd...)*

- After completing transfer, the Program will start to execute.

```
Transfer mode  : UART XMODEM


 IRAM          :  [0x20000 - 0x3AFFF] [108 KB]

 Please send file using XMODEM and then press ENTER key.

 CCCCCCC

C-DAC VEGA Processors

EXIT
```

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Interrupt-Based Example

DIR-V VEGA Processor by C-DAC

Example using interrupts

- Open the example **vega-sdk -> examples -> interrupt -> uart_interrupt**
- Open the **uart_interrupt.c** file
- Inside the main function, update the board frequency to match your board's clock.
- For instance, the TRISUL32 SoC used in this setup runs at 40 MHz.

```c
#define UART0_INTR_NUM 1 // Note : this number is for thejas64, number may change on other SoCs

void uart_intr_handler(void) // interrupt handler
{
    unsigned char status, error, data;

    status = UartReg(UART_0).UART_IIR_FCR; // clearing interrupt
    data = uart_getchar(UART_0, &error);   // reading data
    uart_putchar(UART_0, data, &error);    // echoing data
    return;
}

/**
 @fn main
 @brief transmit and reception through uart
 @details
 @param[in] No input parameter.
 @param[Out] No ouput parameter.
 */
void main()
{

    uart_set_baud_rate(UART_0, 115200, 40000000); // initializing uart0
    printf("+------[ uart%d in interrupt mode ]------+\n", UART_0);

    volatile unsigned int * Bdram = (unsigned int *) (0x20030000);
    * Bdram = (unsigned int) 0x00200002;

    interrupt_enable(UART0_INTR_NUM); //Enable interrupt in controller.
    irq_register_handler(UART0_INTR_NUM, uart_intr_handler);
    uart_intr_enable(UART_0, TX_DISABLE, RX_ENABLE); // enabling uart0 interrupt
    printf("\nStart typing anything on uart%d.\n\n", UART_0);
        /* idle loop */
        while (1);
}
```

Example using interrupts *(contd...)*

- Compile the project, say **uart_interrupt**

```
~/VEGA/vega-sdk/examples/interrupt/uart_interrupt$ make
```

- If your make command end like following,then your build is successful.

```
Build successful!
ELF : uart_interrupt
Binary  : uart_interrupt.bin
Hex : uart_interrupt.hex
Dump    : uart_interrupt.dump
Files are generated in build folder.
Size information
   text    data     bss     dec     hex filename
  30096      36     344   30476    770c build/uart_interrupt.elf
```

Example using interrupts *(contd...)*

- Open minicom :

```
~VEGA/vega-sdk/examples/interrupt/uart_interrupt$ sudo minicom trisul32
```

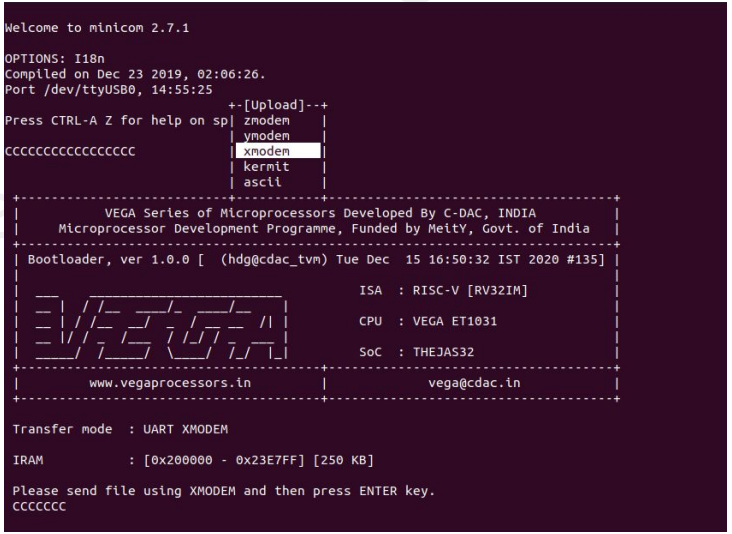
```
/VEGA/vega-sdk/examples/interrupt/uart_interrupt$ sudo minicom trisul32
[sudo] password for xyz:


Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyUSB1, 14:42:08

Press CTRL-A Z for help on special keys
```

- Reset board and make sure Transfer mode is UART XMODEM.

```
+-------------------------------------------------------------------------------+
|          VEGA Series of Microprocessors Developed By C-DAC, INDIA             |
|    Microprocessor Development Programme, Funded by MeitY, Govt. of India      |
|Swadeshi Microprocessor Challenge - Innovate Solutions for #AatmaNirbharBharat|
+-------------------------------------------------------------------------------+
| Bootloader, ver 1.0.0 [  (hdg@cdac_tvm) Mon Jun  30 11:01:32 IST 2025 #160]   |
|                                                                               |
|  ___   ____________________          ISA  : RISC-V [RV32IMA]                  |
|  __ |  / /__  ____/_  ____/__  __          |                                  |
|  __ | / /__  __/  _  /_   __  /| |         CPU  : VEGA AT1051                 |
|  __ |/ / _  /___  / /_/ /  _  ___ |                                           |
|  _____/  /_____/  \____/  /_/  |_|         SoC  : TRISUL32                   |
+-------------------------------------+-----------------------------------------+
|         www.vegaprocessors.in       |             vega@cdac.in                |
+-------------------------------------+-----------------------------------------+
Press any key to choose the xmodem transfer

The ethernet mode will be executed automatically in 03s.

Transfer mode  : UART XMODEM

IRAM           :  [0x20000 - 0x3AFFF] [108 KB]

Please send file using XMODEM and then press ENTER key.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Example using interrupts *(contd...)*

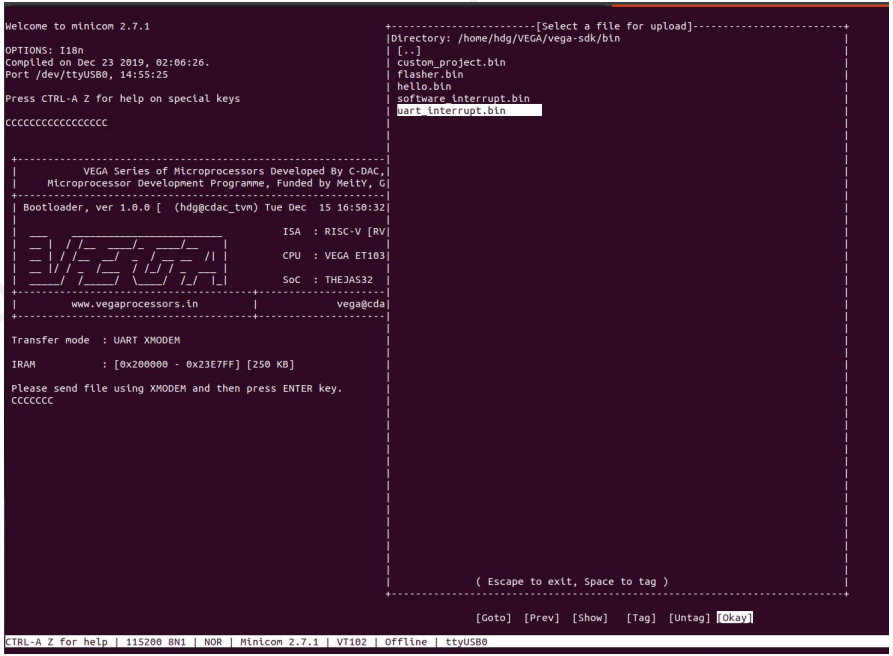
- Use **CTRL+A S** to enter file sending menu and select **xmodem** by pressing Enter.



DIR-V VEGA Proce C-DAC

Example using interrupts *(contd...)*

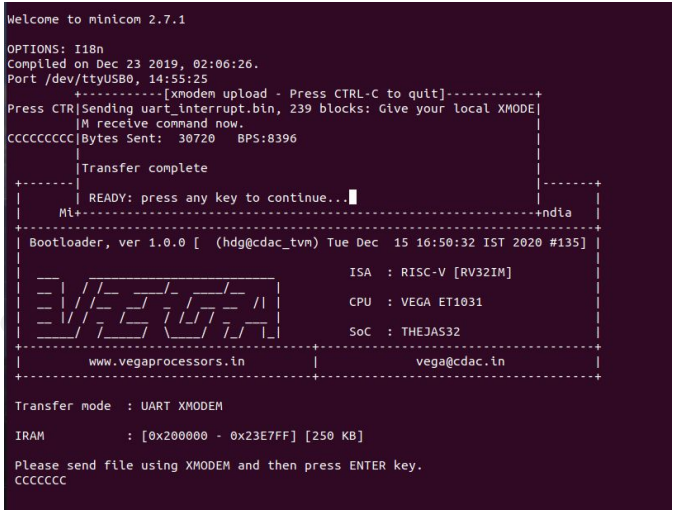
- In the next window, with space bar, select the **.bin** file to be transferred, by pressing **Enter**, transfer process starts.



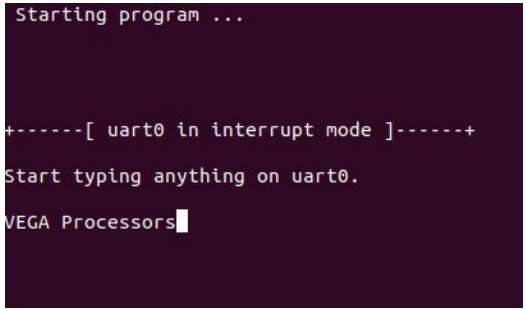
DIR-V VEGA Processor C-DAC

Example using interrupts *(contd...)*

- Wait until the process is completed.
- The screen should display how much data has been transferred.



- After completing transfer the Program will start to execute.



DTR V VEGA Process

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Using SDK in IDE (Eclipse)

DIR-V VEGA Processor by C-DAC

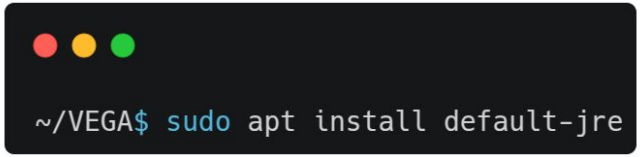
Using SDK in IDE (Eclipse)

Installing Eclipse IDE

Eclipse is a Java-based application and it requires a Java runtime environment (JRE) to be installed in order to run.

Install the default OpenJDK package with:

- `sudo apt install default-jre`



Download Eclipse IDE installer for Linux 64-bit version from https://www.eclipse.org/downloads/

DIR-V VEGA Processor by C-DAC

Using SDK in IDE (Eclipse)(contd...)

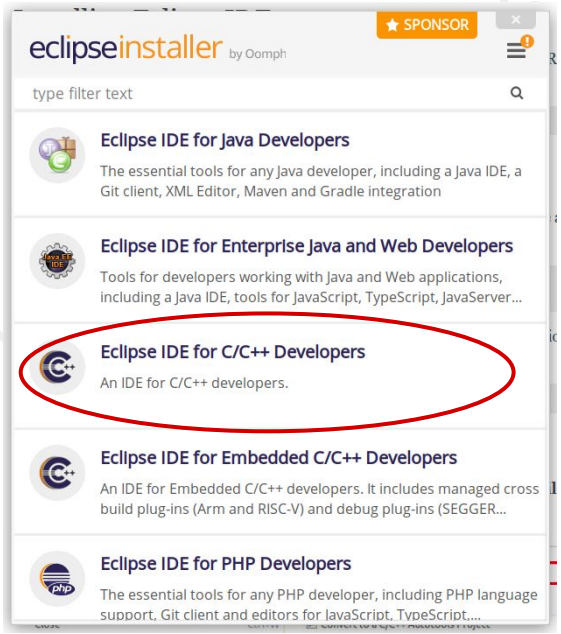
Installing Eclipse IDE

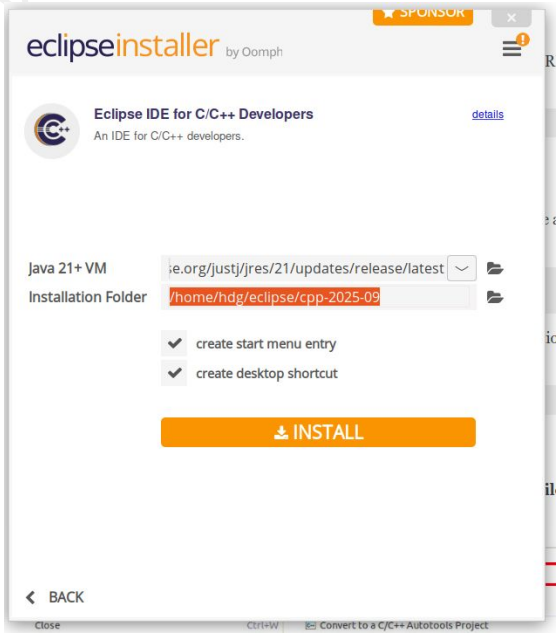
The name of downloaded file will be `eclipse-inst-linux64.tar.gz`

Extract the archive and :

- `cd eclipse-installer`
- `./eclipse-inst`

- Select **Eclipse IDE for C/C++ Developers** and click **INSTALL**.

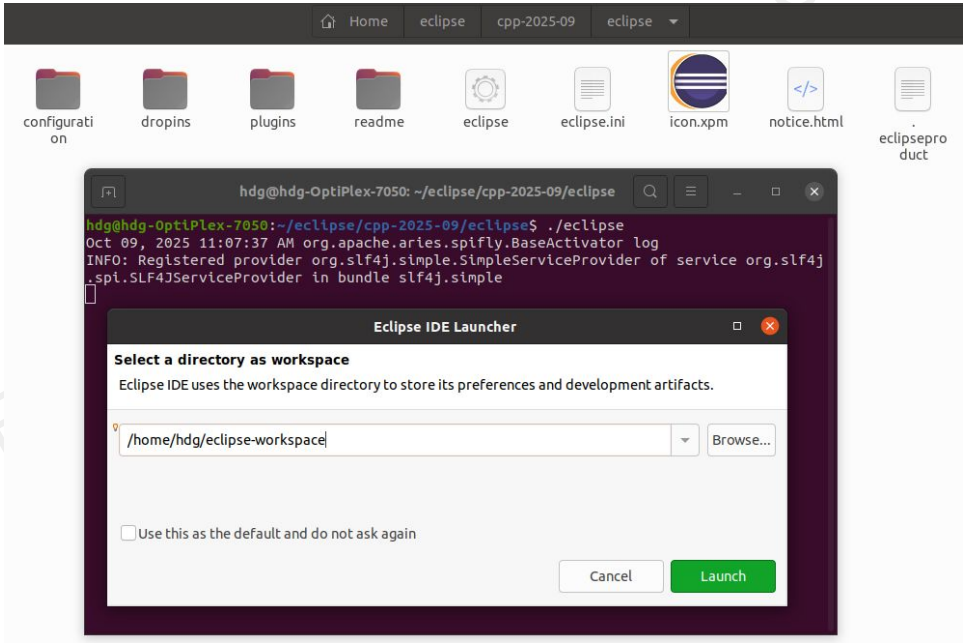




Using SDK in IDE (Eclipse)(contd...)

After installation to Open Eclipse go to installation directory

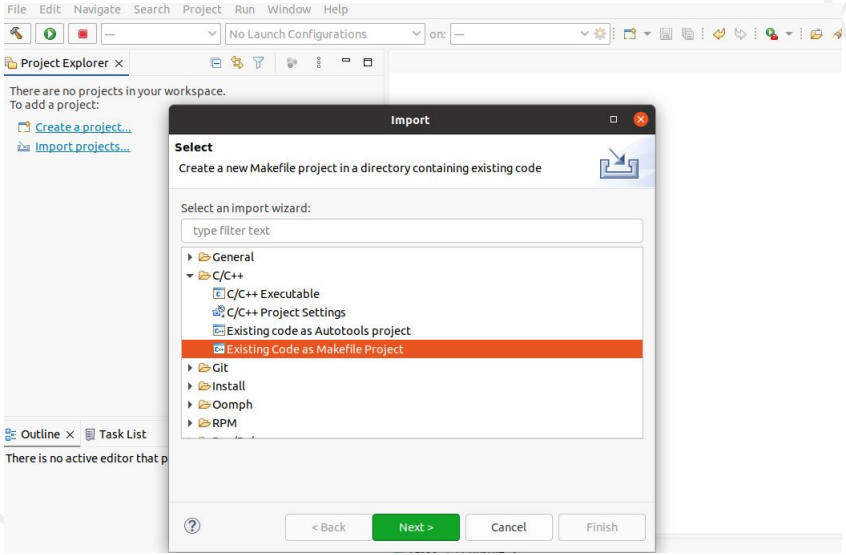
- `./eclipse`



C-DAC
DIR-V VF

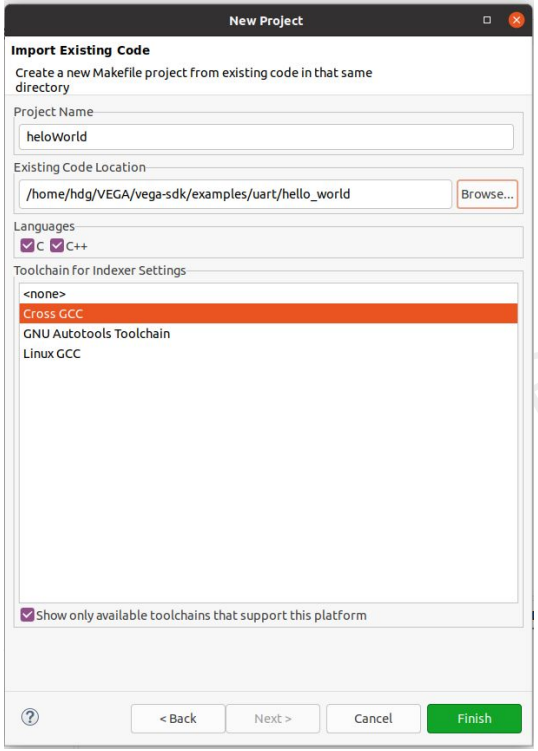
Opening SDK Projects in Eclipse IDE

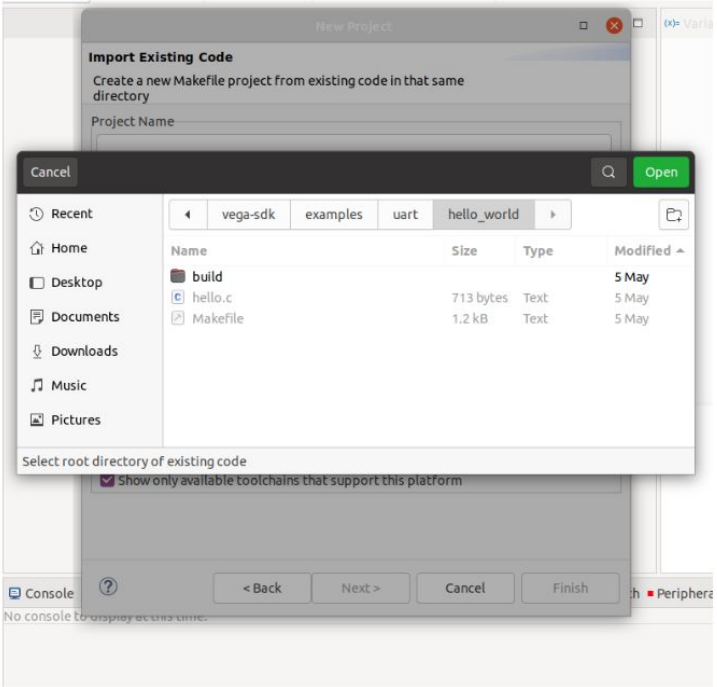
- To open your VEGA SDK project in Eclipse IDE, goto `File -> Import -> C/C++ -> Existing Code as Makefile Project (for 2025 version)`



Opening SDK Projects in Eclipse IDE (contd...)

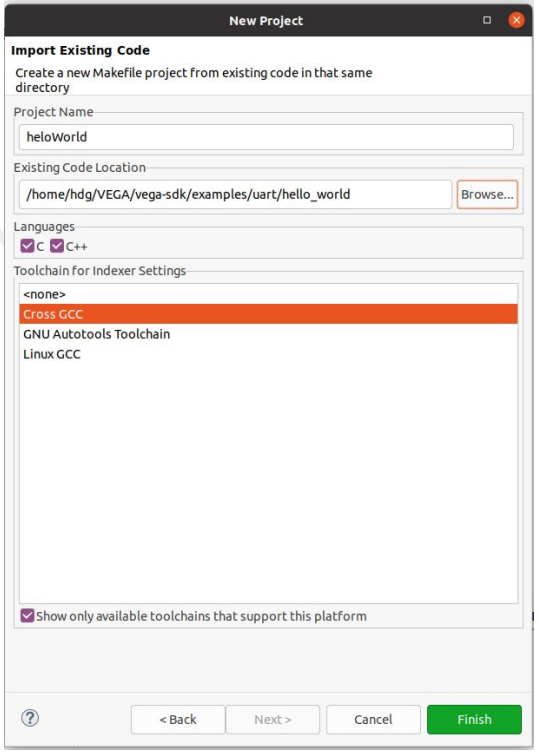
- Browse to existing code location,





Opening SDK Projects in Eclipse IDE (contd...)

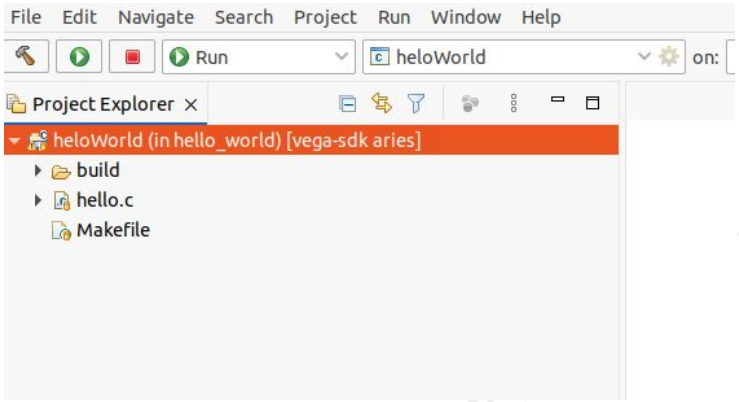
- Select **Cross GCC** option for toolchain indexer and click **Finish** button.

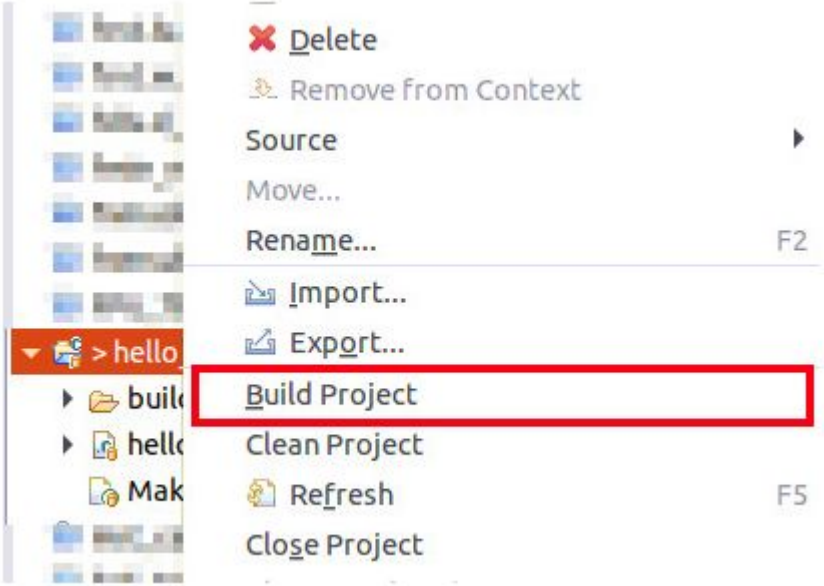


DIR-V VEGA Processor

Building SDK Projects in Eclipse IDE

- In the Project Explorer tab right click and select **Build Project** to build the selected project.

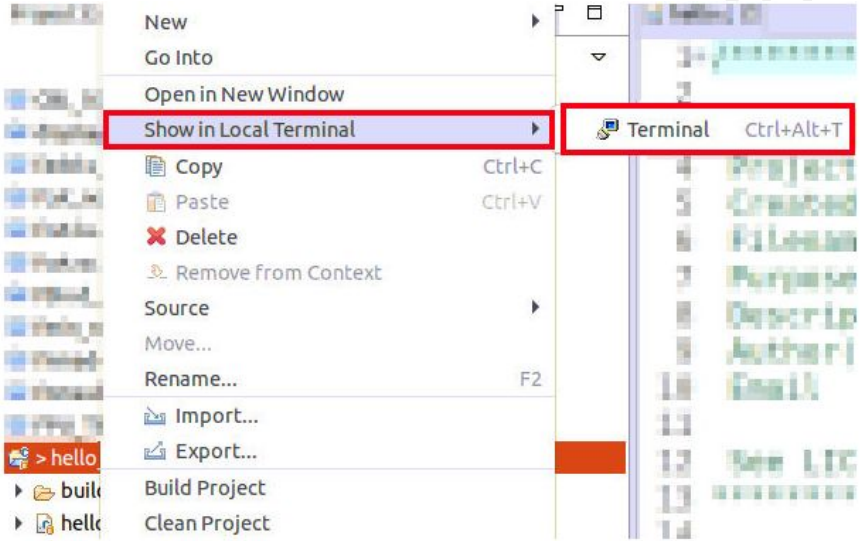




- Selecting **Clean Project** will clean the target program build directory.

Running SDK Projects in Eclipse IDE

- In the Project Explorer tab right click and select **Show in Local Terminal -> Terminal.**



Programming using minicom in Eclipse IDE(contd...)

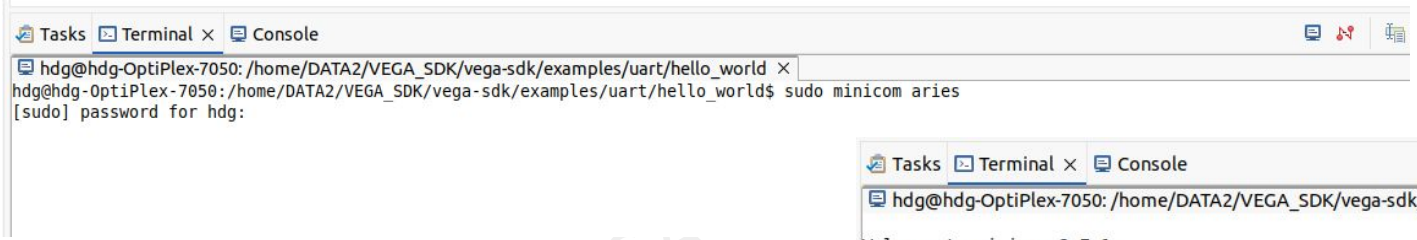
Warning

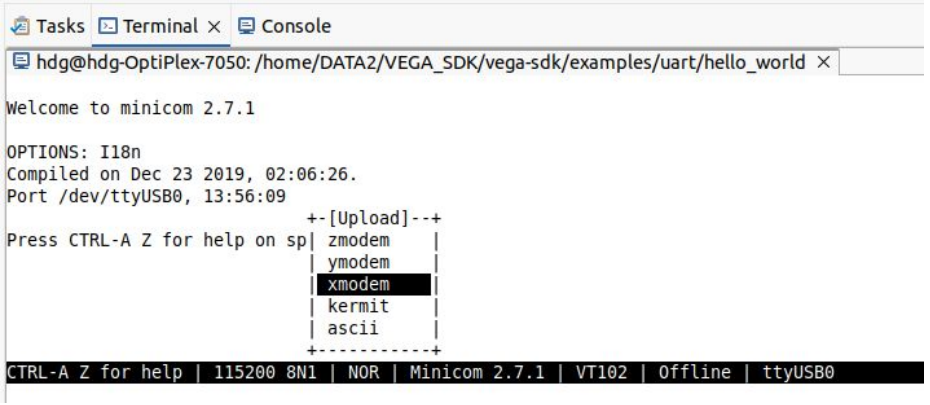
Before giving make upload command do all the steps in Setting Up Serial Device and Resetting Target Board subsections.

Programming using minicom in Eclipse IDE (contd...)

Warning

Before giving make upload command do all the steps in Setting Up Serial Device and Resetting Target Board subsections.

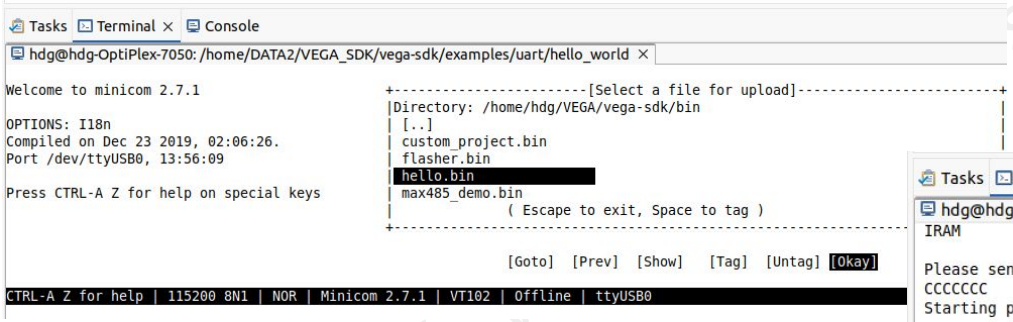


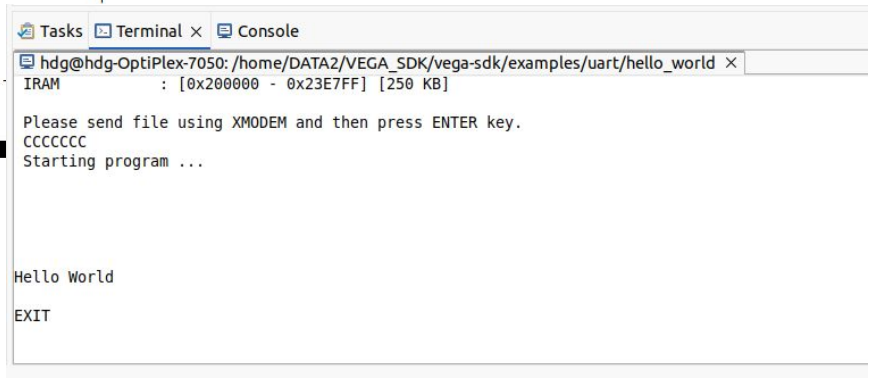


Programming using minicom in Eclipse IDE(contd...)

Warning

Before giving make upload command do all the steps in Setting Up Serial Device and Resetting Target Board subsections.





Warning

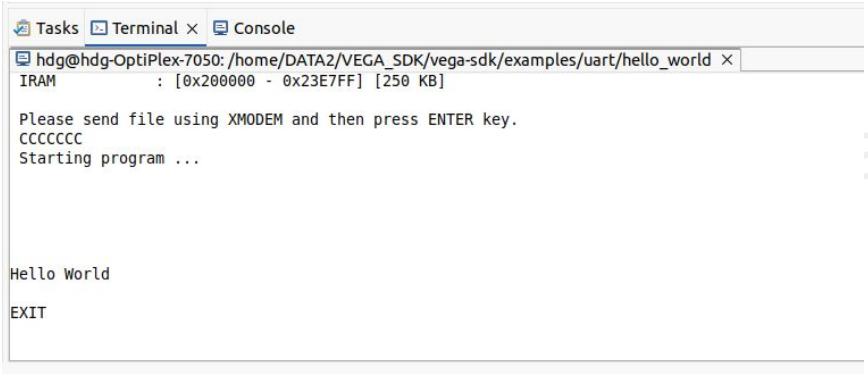
Before giving make upload command do all the steps in Setting Up Serial Device and Resetting Target Board subsections.

```
hdg@hdg-OptiPlex-7050: /home/DATA2/VEGA/vega-sdk/examples/uart/hello_world
hdg@hdg-OptiPlex-7050:/home/DATA2/VEGA/vega-sdk/examples/uart/hello_world$ make upload
TARGET:  THEJAS32 Hardware
C Files : [./hello.c]
CPP Files : []
Assembly Files : []
/home/DATA2/VEGA/vega-tools/toolchain/bin
-march=rv32ima_zicsr -mabi=ilp32 -mcmodel=medany
Changing mode to XMODEM ...Waiting for receiver ping ...done.
Sending /home/DATA2/VEGA/vega-sdk/examples/uart/hello_world/build/hello.bin ....................................
.....................................................................................................done.
#@echo "Please connect the aries board to PC and enter your password to open minicom";
sudo minicom trisul32
[sudo] password for hdg:
```

Programming using make upload in Eclipse IDE (contd...)

In Terminal Tab give command

- `make upload`



Note: Reset the board first, then run the "make upload" command before switching to Ethernet mode; otherwise, you won't get the correct output.

DIR-V GRAND CHALLENGE

VEGA Processor Tutorial

Uploading Using Ethernet

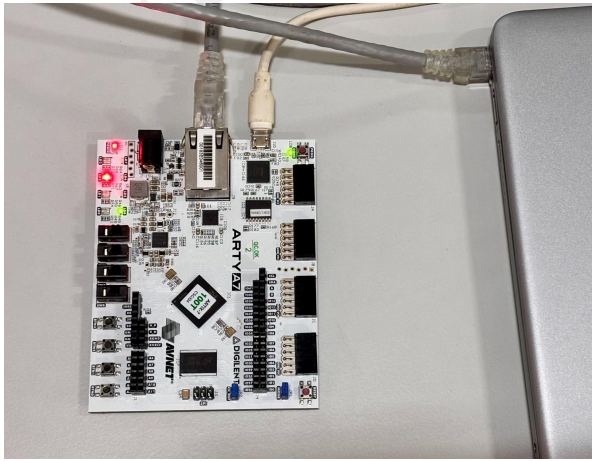
DIR-V VEGA Processor by C-DAC

Uploading Using Ethernet

Make sure that you connected ethernet cable to board and board is reset before you upload program.

Warning

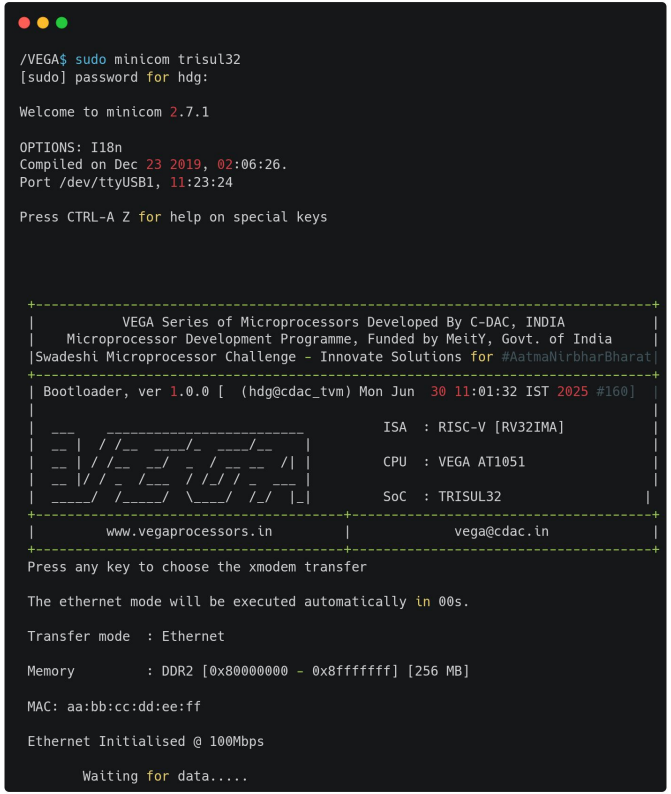
Before giving make upload command do all the steps in Setting Up Serial Device and Resetting Target Board subsections.



DIR-V VEGA Processor by C-DAC

Uploading Using Ethernet *(contd...)*

- After resetting the board, ensure it is in Ethernet mode.
- The board will then wait for incoming data and display the message “waiting for data...” as shown.



DIR-V VEGA Process

Uploading Using Ethernet *(contd...)*

- You could upload program to target board using `make` command.
- To upload program use :

  `make ethupload`

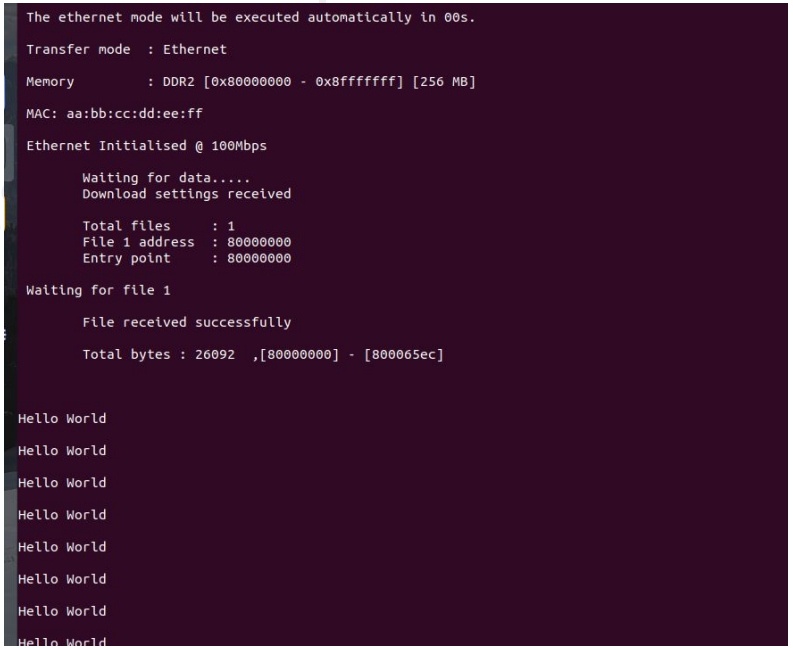
- If the file was uploaded successfully, then you will see the following message in terminal.

```
~/VEGA/vega-sdk/examples/uart/hello_world$ make ethupload
TARGET:  THEJAS32 Hardware
C Files : [./hello.c]
CPP Files : []
Assembly Files : []
/home/hdg/VEGA/vega-tools/toolchain/bin
-march=rv32ima_zicsr -mabi=ilp32 -mcmodel=medany
sudo ./ether enp2s0 /home/hdg/VEGA/vega-sdk/examples/uart/hello_world/build/hello.bin
[sudo] password for hdg:
No of files :1
address 1 : 0x80000000 [ /home/hdg/VEGA/vega-sdk/examples/uart/hello_world/build/hello.bin ]
entry : 0x80000000
entry : 0x20000
Mac : aabbccddeeff
Total files :1
File : /home/hdg/VEGA/vega-sdk/examples/uart/hello_world/build/hello.bin
Read count 26092
File checksum: 246427
```

C-DAC
DIR-V VEGA Pro

Uploading Using Ethernet*(contd...)*

Sample File Upload Output



DIR-V VEGA Process C-DAC

THANK YOU

vega@cdac.in
Mob:9037569219

DIR-V VEGA Processor by C-DAC